RECENT DEVELOPMENTS FOR AN OBSTACLE AVOIDANCE SYSTEM FOR A SMALL AUV

Douglas Horner and Oleg Yakimenko

Naval Postgraduate School
Monterey, CA USA

Abstract: Improvements in high resolution small forward looking sonar (FLS) and computer processing have made it possible to develop an obstacle avoidance system (OAS) for small diameter Autonomous Underwater Vehicles (AUV). An AUV with such a system can maneuver around unanticipated obstacles that may be proud of the ocean floor. This ability can prevent serious damage to the vehicle or the environment. This paper discusses developments in control and computer vision techniques of an OAS designed to vertically avoid obstacles found on the ocean floor. Results are presented from recent in-water testing. *Copyright © 2007 IFAC*

Keywords: Obstacle Avoidance, Optimal Control, Navigation, Obstacle Detection

1. INTRODUCTION

We discuss progress in a developmental OAS mounted on the front of a small AUV. The scope of the problem is limited by considering the avoidance of obstacles proud of the ocean floor. The maneuver around the obstacle is in the vertical plane. In other words, if an obstacle is detected in the path of the AUV, the AUV will maneuver up over the obstacle and return to the pre-planned altitude. Frequently AUVs conduct side scan sonar surveys of the bottom, these surveys are conducted close to the ocean floor to limit gaps in sonar coverage..

The discussion focuses around the image processing and the control necessary to avoid an unanticipated obstacle. This is a continuation of the work first reported in OCEANS 2005 [Horner]. The image processing section focuses on tracking the ocean floor using the forward looking sonar. Using the vehicle state together with a hough space representation of the image, an Extended Kalman Filter (EKF) is constructed to give a predictive estimate where the ocean floor is in the sonar image space [Mills]. Once the ocean floor is reliably tracked, obstacles and features can be tracked that are proud of the ocean floor. When an obstacle is high enough off the ocean floor to be a threat to the vehicle, the pre-planned vehicle path is modified.

A direct method for calculating real-time, near optimal polynomial trajectories [Yakimenko] is presented for the avoidance behavior. The calculation takes into consideration an optimization function based, in part, on the idea of minimizing the altitude change to ensure that sensor coverage stays consistent. The real time trajectory planning has the secondary benefit of being able to react to additional obstacles that may have been occluded by the first identified obstacle.

The paper is organized as follows: Section two describes the obstacle avoidance framework. Section three gives a description of the sonar. Section four discusses sonar image processing. Section five discusses the AUV motion model. Section six discusses optimal problem formulation and Section seven describes near optimal trajectory generation and Section eight describes recent results.

2. FRAMEWORK

The obstacle avoidance framework (Fig. 1) consists of the environmental map, a planning module, a localization module and the sensors and actuators for the control of the robot. The environmental map is the world according to the robot. This can include *a priori* knowledge, for example the positions of charted underwater obstacles. It takes as input the sensor and state information from the robot and translates the sensor information into the global space. The localization module takes as input the most current position estimate from the robot's navigation model and select features from environmental model to provide (if possible) a better estimate of the robots position.

The planning module is responsible for coming up with a trajectory that the robot can follow that is free of obstacles and in some sense optimal. It consists of the deliberative and reactive planners. They take information from the environment and localization modules to produce the trajectory that the robot controller tracks. The deliberative planner uses the environmental map to produce trajectories to a goal state that is free of obstacles. The reactive planner takes recent sensor information to produce a trajectory which will avoid an unanticipated obstacle but provides no guarantees of reaching the end goal state.

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **SEP 2007**	2. REPORT TYPE	3. DATES COVERED **00-00-2007 to 00-00-2007**

4. TITLE AND SUBTITLE **Recent Developments for an Obstacle Avoidance System for a Small AUV**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Postgraduate School,Center on Contemporary Conflict,Monterey,CA,93943**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S)
	11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

IFAC Conference on Control Applications in Marine Systems (CAMS’07), Sep 19-21, Bol, Croatia

14. ABSTRACT

Improvements in high resolution small forward looking sonar (FLS) and computer processing have made it possible to develop an obstacle avoidance system (OAS) for small diameter Autonomous Underwater Vehicles (AUV). An AUV with such a system can maneuver around unanticipated obstacles that may be proud of the ocean floor. This ability can prevent serious damage to the vehicle or the environment. This paper discusses developments in control and computer vision techniques of an OAS designed to vertically avoid obstacles found on the ocean floor. Results are presented from recent in-water testing.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**7**	

3. SONAR DESCRIPTION

The forward looking sonar is a Blazed Array 450 KHz prototype built by Blueview Technologies. The system has a total of six staves, four oriented horizontally and two oriented vertically. Each individual stave has an approximate 23 degrees field of view with 12 degrees of vertical aperture. The range resolution is .092m and the angular resolution is a nominal 1.2 degree. The staves oriented horizontally combine together create a single sonar image. This image has a 90 degree field of view with 12 degrees of vertical ambiguity. The staves oriented vertically combine to produce a single image. This image has a 46 degree field of view and has 12 degrees of horizontal ambiguity. For this paper, only the vertical images are considered. Figure 2 is an example of a vertical image.

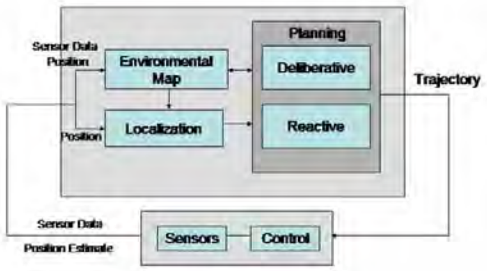

Figure 1. Obstacle Avoidance Framework

4. SONAR IMAGE PROCESSING

This section describes some of the techniques used to detect and track an obstacle using the vertically mounted FLS arrays.

4.1 Detection of the Ocean Floor

The primary technique used to detect the ocean floor is the Hough Transform [Hough]. It is a common technique in image processing for the identification of lines from a set of 2D points [Duda and Hart]. In mobile robotics, the Hough Transform (HT) has been used for a range of applications including vision based self-localization [Iocchi and Nardi]. The HT uses the parameterized line equation

$$\rho = x\cos\theta + y\sin\theta \qquad (1)$$

where θ is the angle of the line to the x-axis, ρ is the perpendicular distance from the line and the origin of a selected point in the image space, and x and y represent a point on the line. A transformation from the original image to a hough image converts the (x, y) image points into the (ρ,θ) hough space. A line in the image space is represented as a point in the hough space. In this case, detection of the ocean floor is accomplished by searching for a point in the hough space.

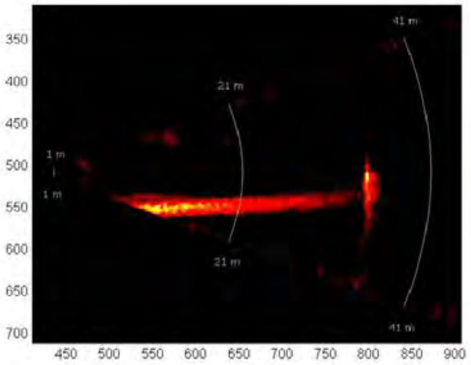

Figure 2. Example of a vertical sonar image with a obstacle located approximately 35M from the AUV

Consistent tracking of the ocean floor is made easier by taking into account the state of the vehicle and using this information to track the position of the ocean floor in the hough image space. Specifically, an EKF has been used to estimate where the point (i.e. the ocean floor) should be in the hough space given the current AUV position and a motion model.

The EKF takes as input AUV pitch, roll, heading, heading rate, surge, and heave, altitude and ocean floor point in the hough space (ρ,θ) and produces a predictive estimate of ocean floor in the hough space. This is converted back into the image space and is used as the reference line in the search for obstacles.

4.2 Detection of Obstacles

In previous work, using a model for the motion of the REMUS AUV an obstacle was determined to be a collision threat to the AUV when the leading edge slope of the obstacle is greater than a nominal 45 degree angle to the ocean floor. This is the angle at which the Doppler Velocity Log (DVL) reporting at 2Hz can no longer provide altitude data quickly enough for the vehicle control to overcome the altitude change. This is used as the template for determining when to activate an avoidance behavior.

With each image, the ocean floor is first localized. Next a search box is constructed for the relevant area forward of the AUV. The area is searched using the cvBlobsLib library[1]. When an object is detected a threshold number of times and is a threat to the vehicle it is classified as an obstacle.

The Hough EKF also helps in separating out obstacles in the sonar image. By tracking the ocean floor in the hough space obstacles can be separated from the ocean floor by the angle of the obstacle in the image space and the distance from the image center ρ. Figure 3 shows the original image (from Figure 2) in the hough space. Notice the obstacle is visible as a separate curved line where the peak

[1] http://opencvlibrary.sourceforge.net/cvBlobsLib

intensity is at the approximate coordinate $(\rho = 111, \theta = 179)$.

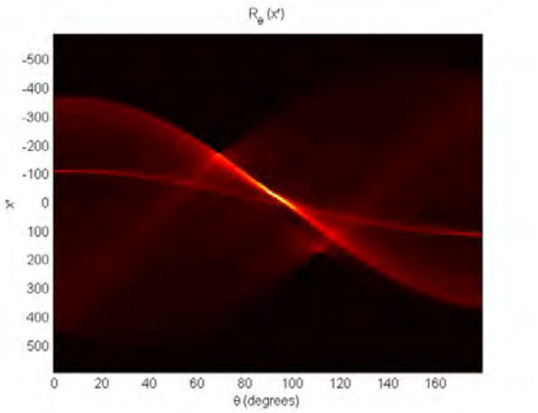

Figure 3. Hough representation of the vertical sonar image shown above.

5. AUV MODEL

This section describes the AUV model used by the reactive trajectory planner. Consider the two-dimensional kinematics equation of motion for the vertical channel for the AUV.

$$\begin{aligned}\dot{x} &= w\sin\theta + u\cos\theta \\ \dot{z} &= w\cos\theta - u\sin\theta\end{aligned} \qquad (2)$$

where *x* is the AUV longitudinal position, *z* is its altitude with respect to the ocean floor, u and w are the forward and heave velocities respectfully and θ is the pitch angle**.** Next consider the dynamics equation for the REMUS AUV for vertical motion using the standard state-space representation with the state vector $\mathbf{x}_\mathrm{v}(t) = [w, q, \theta]^T$:

$$\dot{\mathbf{x}}_\mathrm{v}(t) = \mathbf{A}_\mathrm{v}\mathbf{x}_\mathrm{v}(t) + \mathbf{B}_\mathrm{v}\delta_s(t) \qquad (3)$$

Here q is the pitch rate and δ_s is the dive plane deflection, the matrices **A** and **B** (4) are the multiplication of the inverse of the mass matrix with the motion model. The derivation of the matrices **A** and **B** and the coefficients used for the forces, moments and mass moments of inertia are taken from [Healey].

$$\begin{aligned}\mathbf{A}_\mathrm{v} &= \begin{bmatrix} m-Z_{\dot{w}} & -Z_{\dot{q}} & 0 \\ -M_{\dot{w}} & I_{yy}-M_{\dot{q}} & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} Z_w & mu_0+Z_q & 0 \\ M_w & M_q & z_B B - z_G mg \\ 0 & 1 & 0 \end{bmatrix} \\ \mathbf{B}_\mathrm{v} &= \begin{bmatrix} m-Z_{\dot{w}} & -Z_{\dot{q}} & 0 \\ -M_{\dot{w}} & I_{yy}-M_{\dot{q}} & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} Z_{\delta_s} \\ M_{\delta_s} \\ 0 \end{bmatrix}\end{aligned} \qquad (4)$$

6. OPTIMAL PROBLEM FORMULATION

The next step is to formulate the general optimization problem as applied to the problem at hand. We post it as follows.

There is set of admissible trajectories

$$\begin{aligned}&\mathbf{z}(t) = \left[x(t), z(t), w(t), q(t), \theta(t)\right]^T \in S, \\ &S = \left\{\mathbf{z}(t) \in Z^5 \subset E^5\right\},\ t \in \left[t_0, t_f\right]\end{aligned} \qquad (5)$$

that satisfies a set of ordinary differential equations (2)-(3)

$$\dot{\mathbf{z}}_i = \mathbf{f}_i(\mathbf{z}, \delta_\mathrm{s}, \mathbf{c}) \qquad i = 1, 2, \ldots, 5$$

where $\delta_\mathrm{s}(t)$ is the only control and $\mathbf{c} = \{c_1, c_2, \ldots, c_p,\}, \mathbf{c} \in C^P$ is the vector of AUV, characteristics with initial conditions $(\mathbf{z}(t_0), \delta_\mathrm{s}(t_0))$, terminal conditions $(\mathbf{z}(t_f), \delta_\mathrm{s}(t_f))$ and the following constraints on state space controls and control derivatives (listed in order):

$$\begin{aligned}&\boldsymbol{\eta}(t,\mathbf{z}) = \{\eta_1(t,\mathbf{z}), \eta_2(t,\mathbf{z}), \ldots, \eta_w(t,\mathbf{z})\}^T \geq \mathbf{0} \\ &\boldsymbol{\xi}(t,\mathbf{z}) = \{\xi_1(t,\mathbf{z},\delta_\mathrm{s}), \xi_2(t,\mathbf{z},\delta_\mathrm{s}), \ldots, \xi_v(t,\mathbf{z},\delta_\mathrm{s})\}^T \geq \mathbf{0} \\ &\boldsymbol{\pi}(t,\mathbf{z}) = \{\pi_1(t,\mathbf{z},\delta_\mathrm{s}), \pi_2(t,\mathbf{z},\delta_\mathrm{s}), \ldots, \pi_v(t,\mathbf{z},\delta_\mathrm{s})\}^T \geq \mathbf{0}\end{aligned} \qquad (6)$$

The problem is to find an optimal trajectory and optimal control $\delta_{sopt}(t)$ that minimizes some performance index, for instance some integral index

$$J = \int_{t_0}^{t_f} f_0(t, \mathbf{z}, \delta_\mathrm{s})dt \qquad (7)$$

For an AUV constraints (8) result in 1) vertically avoiding the obstacle, 2) staying within the constraints on the dive plane deflection δ_s, and 3) accounting for plane deflection dynamics (modeling as a first-order system):

$$\begin{aligned}\eta(\mathbf{z}) &= z(x_{obs}) - z_{obs} > 0 \\ \xi(\delta_s) &= \delta_{s\max} - \left|\delta_s(t)\right| \geq 0 \\ \pi(\dot{\delta}_s) &= \dot{\delta}_{s\max} - \left|\dot{\delta}_s(t)\right| \geq 0\end{aligned} \qquad (8)$$

As a performance index (9) it is natural to use the combined one that 1) minimizes the distance traveled to avoid the obstacle, and 2) maximizes the amount of time with the pitch of the AUV between nominal θ_{nom} of 0 and -3 degrees:

$$J = \int_{t_0}^{t_f} \left[(z(t) - z_0)^2 + k(\theta(t) - \theta_{nom})^2\right]dt \qquad (9)$$

where *k* is some scaling coefficient (of the order of 11). This later one is to try and ensure the sonar is looking forward a maximum amount of time during the avoidance maneuver.

7. NEAR OPTIMAL CONTROL FOR REACTIVE OBSTACLE AVOIDANCE

When an obstacle is identified by the image processing routine, an OA behavior is activated and the position and maximum height of the obstacle is sent to this process. Originally both with the NPS ARIES and REMUS AUVs reactive OA behavior was achieved by applying an additive Gaussian altitude curve over the obstacle position. This proved to be an adequate first solution but had two detractions. First, any obstacle proud of the ocean floor ensonified by the FLS, may occlude any objects behind it. Using a Gaussian additive altitude approach the AUV runs the risk of navigating into another obstacle on the “backside” of the Gaussian curve. Second, the implementation doesn’t permit an AUV to react to moving objects.

Instead the approach is to use near optimal real-time polynomial trajectory path planner for reactive obstacle avoidance. The trajectory generation and optimization procedures [Yakimenko] are now described for AUV navigation. This direct-method-based procedure assures the following:

- the boundary conditions including high-order derivatives are satisfied a priori,
- the control commands are smooth and physically realizable,
- the method is very robust and is not sensitive to small variations in the input parameters;
- only a few variable parameters are used, thus ensuring that the iterative process during optimization converges well and that the continuous update of the solution allows reliable path following even with no standard feedback.

Another important feature of this method is that it allows handling any compound performance index, like in (11). Therefore, we are not limited to simple standard indices like time, fuel expenditure, etc.

7.1 Generating a Candidate Trajectory

Consider a desired AUV trajectory as $p(\tau)=[x(\tau),z(\tau)]^T$ where τ is some abstract argument, the virtual arc, and is defined by $\tau=[0;\tau_f]$, where τ_f is the total length of the virtual arc. Each coordinate x_i, $i=1,2$ (for simplicity of notation we assume $x_1(\tau)\equiv x(\tau)$ and $x_2(\tau)\equiv z(\tau)$) is represented by a polynomial of degree N with the form:

$$x_i(\tau)=\sum_{k=0}^{M} a_{ik}^{*}\tau^k, \qquad i=1,2 \qquad (10)$$

The degree of the polynomial M is determined by the number of boundary conditions that must be satisfied. In our case the desired trajectory includes constraints on the initial and final position, velocity and acceleration. In this case the minimal order of the polynomial is 5. However, to increase the number of varied parameters and allow for more flexibility of the candidate trajectory the order of polynomials can be higher than 5. Say we go with the seventh-order polynomial, so that

$$\begin{aligned}
x_i'''(\tau) &= \sum_{k=3}^{7} a_{ik}\tau^{k-3}\\
x_i''(\tau) &= \sum_{k=2}^{7} \frac{1}{\max(1,k-2)} a_{ik}\tau^{k-2}\\
x_i'(\tau) &= \sum_{k=1}^{7} \frac{1}{\max(1,(k-1)(k-2))} a_{ik}\tau^{k-1}\\
x_i(\tau) &= \sum_{k=0}^{7} \frac{(\max(1,k-3))!}{k!} a_{ik}\tau^{k},
\end{aligned} \qquad (11)$$

The coefficients a_{ik}, $i=1,2$, $k=0,...,7$ can be determined by solving the following system of linear algebraic equations

$$\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\
1 & \tau_f & \frac{1}{2}\tau_f^2 & \frac{1}{6}\tau_f^3 & \frac{1}{24}\tau_f^4 & \frac{1}{60}\tau_f^5 & \frac{1}{120}\tau_f^6 & \frac{1}{210}\tau_f^7\\
0 & 1 & \tau_f & \frac{1}{2}\tau_f^2 & \frac{1}{6}\tau_f^3 & \frac{1}{12}\tau_f^4 & \frac{1}{20}\tau_f^5 & \frac{1}{30}\tau_f^6\\
0 & 0 & 1 & \tau_f & \frac{1}{2}\tau_f^2 & \frac{1}{3}\tau_f^3 & \frac{1}{4}\tau_f^4 & \frac{1}{5}\tau_f^5\\
0 & 0 & 0 & 1 & \tau_f & \tau_f^2 & \tau_f^3 & \tau_f^4
\end{pmatrix}
\begin{pmatrix} a_{i0}\\ a_{i1}\\ a_{i2}\\ a_{i3}\\ a_{i4}\\ a_{i5}\\ a_{i6}\\ a_{i7} \end{pmatrix}
=
\begin{pmatrix} x_{i0}\\ x_{i0}'\\ x_{i0}''\\ x_{if}'''\\ x_{if}\\ x_{if}'\\ x_{if}''\\ x_{if}''' \end{pmatrix} \qquad (12)$$

As seen, no matter what value of the final arc τ_f we have, the boundary conditions will be unconditionally satisfied. But varying this value allows varying the shape of the candidate trajectory. Figure 4 demonstrates an example when an AUV navigates at 3m altitude at 1.5m/s and in order to avoid some obstacle has to perform a pop-up maneuver. Changing the value of τ_f allows avoiding obstacles with different height.

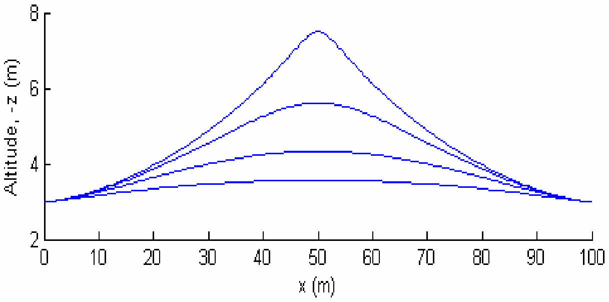

Figure 4. Variances of the candidate trajectory while changing the value of τ_f.

Now what about four free parameters which in our case are components of the initial and final jerk, x_{i0}'''

and x'''_{if}, $i = 1, 2$ (15)? Having these additional varied parameters allows to further change the overall shape of the trajectory to be able to vary the combined performance index (11) we chose. To this end Fig.5 shows an example of an AUV navigating at 3m altitude at 1.5m/s and trying to avoid a 6m obstacle located in between initial and final points.

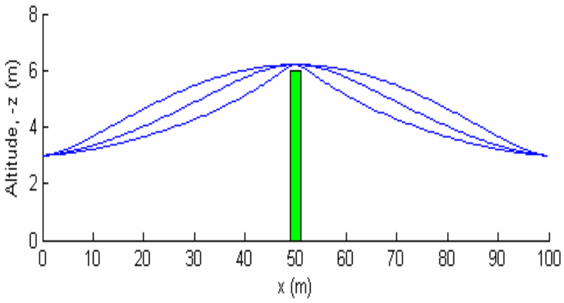

Figure 5. Symmetric varying of x'''_{10} and x'''_{1f} to avoid an obstacle with optimizing τ_f.

Now, let us address the reason for choosing some abstract parameter τ (not time, not path length) as an argument for the reference functions (12)-(13). Assume for a moment that $\tau \equiv t$. In this case once we defined the trajectory we unambiguously defined the speed profile along this trajectory as well since

$$V(t) = \sqrt{u^2 + w^2} = \sqrt{\dot{x}(t)^2 + \dot{z}(t)^2} \qquad (13)$$

Obviously we cannot allow this and would want to vary the speed profile independently. With an abstract argument it becomes possible via introducing the speed factor λ such that

$$\lambda(\tau) = \frac{d\tau}{dt} \qquad (14)$$

Now instead of (16) we have

$$V(\tau) = \lambda(\tau)\sqrt{x'(\tau)^2 + z'(\tau)^2} \qquad (15)$$

and by varying $\lambda(\tau)$ can achieve any speed profile we want. Therefore using a virtual arc becomes a key element in the proposed approach.

All together the robustness of the approach allows generating the trajectory to accommodate for sudden changes like newly discovered obstacles. As an example Fig. 6 demonstrates the scenario when an AUV avoiding the first obstacle discovers that there is another one right behind the first and to avoid the collision a new trajectory starting exactly with the current states and control (up to the second-order derivatives of the states) needs to be generated. The method allows doing this and assures smooth non-shock transition.

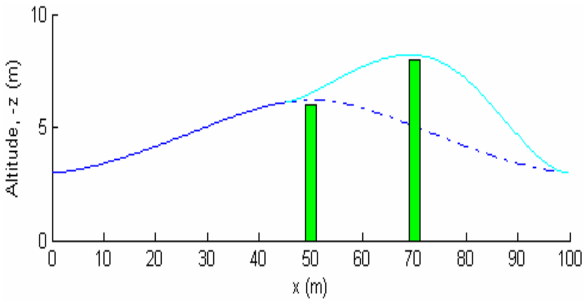

Figure 6. Example of the dynamic reconfiguration of the trajectory.

7.2 Inverse Dynamics

Having the candidate trajectory determined by (13)-(15) the goal now is use inverse dynamics with the AUV model to calculate the remaining states and dive plane commands that is required to follow this candidate trajectory. In this way we can ensure that the candidate trajectory does not exceed vehicle or configuration space constraints. Using the relation (16) for any parameter ζ,

$$\dot{\zeta}(\tau) = \frac{d\zeta}{d\tau}\frac{d\tau}{dt} = \zeta'(\tau)\lambda(\tau) \qquad (16)$$

and AUV motion model the it is possible to convert the equations of motion into the τ domain

$$\begin{array}{ll}
\dot{x} = w\sin\theta + u\cos\theta & x' = \lambda^{-1}(w\sin\theta + u\cos\theta) \\
\dot{z} = w\cos\theta - u\sin\theta & z' = \lambda^{-1}(w\cos\theta - u\sin\theta) \\
\dot{\theta} = q & \theta' = \lambda^{-1}q \\
\dot{w} = A_{11}w + A_{12}q + A_{13}\theta + B_1\delta_s & w' = \lambda^{-1}(A_{11}w + A_{12}q + A_{13}\theta + B_1\delta_s) \\
\dot{q} = A_{21}w + A_{22}q + A_{23}\theta + B_2\delta_s & q' = \lambda^{-1}(A_{21}w + A_{22}q + A_{23}\theta + B_2\delta_s)
\end{array} \qquad (17)$$

We will further use a slightly modified scheme and having the boundary conditions needed to be satisfied and some initial guesses on the varied (free) parameters we only establish a reference polynomial for the vertical coordinate $z(\tau)$ (13). Now we proceed with a numerical procedure for computing all remaining states along the reference trajectory over a fixed set of N points (for instance, N=100) equidistantly placed along the virtual arc $[0;\tau_f]$ with the interval of

$$\Delta\tau = \frac{\tau_f}{N-1} \qquad (18)$$

We take the very next value for the virtual arc

$$\tau_j = \tau_{j-1} + \Delta\tau \qquad j = 2, \ldots, N \qquad (19)$$

and compute the corresponding parameters of the trajectory from $p_z(\tau)$ and $p_{z'}(\tau)$ (13). Having these, we proceed with computing the corresponding time period Δt_{j-1} and elapsed time t_j:

$$\begin{aligned}\Delta t_{j-1} &= t_j - t_{j-1} \approx \frac{z_j - z_{j-1}}{w_{j-1}} \\ t_j &= t_{j-1} + \Delta t_{j-1} \;\; (t_1=0)\end{aligned} \qquad (20)$$

Next we compute the current value of the speed factor:

$$\lambda_j = \frac{\Delta\tau}{\Delta t_{j-1}} \;\; (\lambda_1 = u_0 = 1.5) \qquad (21)$$

Having those we can exploit two of the equations in the virtual domain (20) as follows:

$$\begin{aligned} x'_{j-1} &= \lambda_{j-1}^{-1}(w_{j-1}\sin\theta_{j-1} + u_0\cos\theta_{j-1}) \\ w'_{j-1} &= \lambda_{j-1}^{-1}\left(A_{11}w_{j-1} + A_{12}q_{j-1} + A_{13}\theta_{j-1} + B_1\delta_{s;j-1}\right) \\ x_j &= x_{j-1} + \Delta\tau x'_{j-1} \\ w_j &= w_{j-1} + \Delta\tau w'_{j-1} \end{aligned} \qquad (22)$$

The next step is to compute the pitch angle, pitch rate and pitch acceleration:

$$\begin{aligned} \theta_j &= \cos^{-1}\left(\lambda_j \frac{u_0 x'_{j-1} + w_j z'_j}{w_j^2 + u_0^2}\right) \\ q_j &= \dot{\theta}_j \approx \frac{\theta_j - \theta_{j-1}}{\Delta t_{j-1}} \\ \dot{q}_j &= \ddot{\theta}_j \approx \frac{q_j - q_{j-1}}{\Delta t_{j-1}} \end{aligned} \qquad (23)$$

Finally, we compute the dive plane deflection required to follow the trajectory

$$\delta_{s;j} = \frac{1}{B_2}(\dot{q}_j - A_{21}w_j - A_{22}q_j - A_{23}\theta_j) \qquad (24)$$

7.3 Optimization

When all parameters (states and control) are computed in each of N points, we can compute the performance index and the penalty function. Specifically, according to (10), the penalty function can be formed as follows:

$$\Delta = \left[1, k^{\delta}, k^{\dot{\delta}}\right] \begin{bmatrix} \min\left(0; z(x_{obs}) - z_{obs}\right)^2 \\ \max\limits_j\left(0; \left|\delta_{s;j}\right| - \delta_{s\max}\right)^2 \\ \max\limits_j\left(0; \left|\dot{\delta}_{s;j}\right| - \dot{\delta}_{s\max}\right)^2 \end{bmatrix} \qquad (25)$$

where the penalties on violation of constraints on controls and their derivatives are scaled with the corresponding factors. Now the problem can be solved say in the MATLAB development environment with the built-in *fmincon* function or *fminsearch* function. In the latter case we have to combine the performance index and the penalty function together.

8. RESULTS

Demonstrations were recently completed in an exercise sponsored by the Office of Naval Research in Panama City FL. The obstacle was a sunken 60m oil supply ship. At its highest point the vessel was approximately 12 meters off the ocean floor. Runs were made at the vessel perpendicular to the direction of the keel. The optimal trajectory planner was not implemented on these runs. Figure 7 shows a side scan sonar image which shows the response of the AUV to the sunken ship.

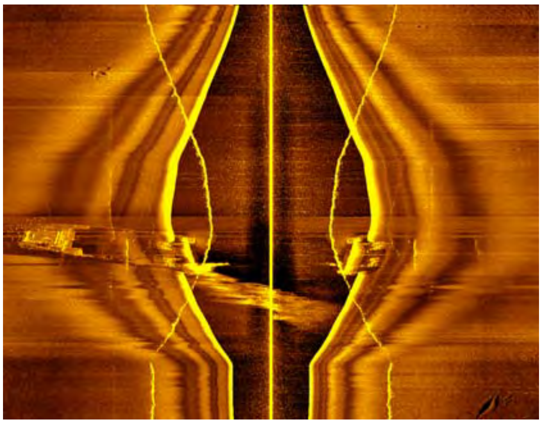

Figure 7. Side scan sonar image from the AUV during an reactive avoidance manuever.

REFERENCES

Duda, R.O. and Hart, P.E., (1972) Use of the Hough transform to detect lines and curves in pictures. Communications of the ACM, 15(1):11-15.

Healey, A. J.,Obstacle Avoidance While Bottom Following for the REMUS Autonomous Underwater Vehicle. Proceedings of the IFAC Conference, Lisbon, Portugal, July 5-7, 2004.

Horner D.P., Healey, A..J., Kragelund, S.K. (2005) AUV Experiments in Obstacle Avoidance. . Proceedings of the IEEE Oceans 2005 Conference, September 18-23, 2005, Washington DC.

Hough P. (1962) Method and means for recognizing complex patterns, U.S. Patent No. 3,069,654. Technical report, HelpMate Robotics Inc.

Iocchi, L. and Nardi, D. (1999). Hough transform based localization for mobile robots. *Advances in Intelligent Systems and Computer Science*, (Mastorakis, N. editor) pages 359.364. World Scientific Engineering Society.

Mill,S., Pridmore, T., Hills, M. Tracking in a Hough Space with the Extended Kalman Filter. Visual Information Engineering, 2003. VIE 2003. Volume , Issue , 7-9 July 2003 Page(s):53-56.

Yakimenko, O. Direct Method for Rapid Prototyping of Near Optimal Aircraft Trajectories. AIAA Journal of Guidance, Control, and Dynamics, 23(5): 865-875, 2000.